МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М. В. Ломоносова

Механико - Математический факультет
Кафедра теории вероятностей

КУРСОВАЯ РАБОТА

«Применение семплирования по Гиббсу к задаче о восстановлении многомерного распределения»

Выполнила студентка группы 308
кафедры теории вероятностей
Меркушина А. В.

Научный руководитель:
доктор физико-математических наук, профессор
Яровая Е. Б.

Москва, 2020

Содержание

Введение

Обработка и анализ данных, имеющих пропущенные значения, являются актуальным вопросом прикладной математической статистики. Например, выборки, собранные в больших эпидемиологических исследованиях, зачастую содержат пропущенные значения [1]. При предварительной обработке данных исследователь вынужден либо восстановить пропуск в наблюдениях, либо удалить участника, имеющего потерянное значение. Замена средним, медианой или константой может оказаться некорректной [2]. Также удаление всех наблюдений, связанных с утерянным, часто недопустимо из-за того, что объём выборки может значительно сократиться. По этим причинам в последнее десятилетие развились подходы, задача которых — восстановить совместное распределение переменных выборки и подобрать для пропущенного значения наиболее правдоподобное заполнение [2].

Цель работы — изучить теоретическую основу, необходимую для применения метода восстановления пропущенных данных. Для этого в главе 1 сделан краткий обзор метода Монте-Карло, на основе которого будет построен метод восстановления пропущенных значений. В главе 2 приведены необходимые сведения из теории однородных марковских цепей с дискретным числом состояний и марковских процессов с непрерывным множеством состояний, сформулирована эргодическая теорема. В главе 3 приведен алгоритм гиббсовского семплирования [6] — один из методов Монте-Карло по схеме марковской цепи. Этот метод является одним из теоретических обоснований алгоритма заполнения пропущенных данных. Кроме того, в главе три нами предложен пример применения Гиббсовского семплирования к задаче моделирования выборки из многомерной случайной величины, которую нельзя решить с помощью функциональных преобразований равномерной случайной величины.

1 Метод Монте-Карло

В этой главе мы рассмотрим метод Монте-Карло, широко используемый в статистике.

Модельный пример метода — подсчёт числа π. Заполним квадрат точками со случайными координатами, распределёнными по двумерному равномерному закону. Рассчитаем отношение количества точек, попавших в круг, к общему количеству точек. Вероятность попадания точки в круг равна отношению площадей круга и квадрата:

$$\frac{\pi R^2}{(2R)^2} = \frac{\pi}{4} \simeq \frac{N_0}{N_1}. \tag{1}$$

Чем больше количество точек, тем ближе полученное значение к значению числа π (см. Рис.1).

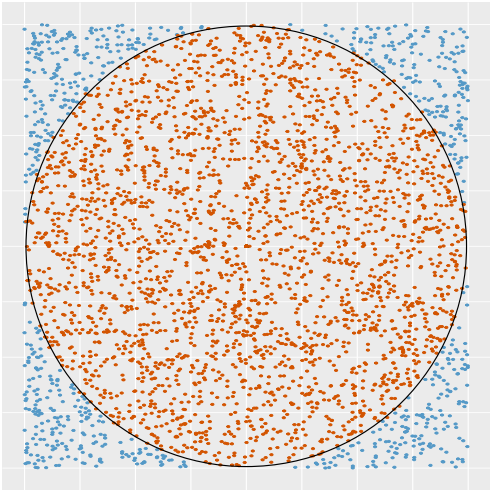

Рис. 1. Распределение точек при вычислении числа π методом Монте-Карло.

Метод Монте-Карло был, по-видимому, впервые предложен в работе [3] для вычисления интегралов посредством генерации случайных величин. Предположим, мы хотим провести вычисление интеграла, аналитическая реализация которого трудоемка. В таком случае часто используются методы численного интегрирования. Для примера рассмотрим метод прямоугольников:

$$I = \int_a^b f(x)dx \cong \frac{(b-a)}{N} \sum_{k=1}^{N} f(\hat{x}_k), \tag{2}$$

где x_k — разбиение отрезка $[a, b]$ на равные части, а $\hat{x}_k = \frac{(x_{k+1}-x_k)}{2}$ — произвольно отмеченные точки.

В многомерных задачах численные методы становятся неприемлимыми из-за того, что количество точек разбиений растет как N^d, где d—размерность пространства. В этом случае можно применить метод Монте-Карло.

В методе Монте-Карло вместо реализации (2) предлагается взять

$$\hat{I} = \frac{(b-a)}{N} \sum_{k=1}^{N} f(X_k), \tag{3}$$

где X_k — н.о.р. $U[a, b]$ [1].

Применяя к $f(X_k)$ ЗБЧ, видим, что $\hat{I}$ сходится к

$$(b-a)\mathbb{E}f(X_k) = (b-a)\int_a^b f(x)\frac{1}{(b-a)}dx = I. \tag{4}$$

Таким образом, для пространств больших размерностей количество точек, в которых необходимо посчитать значения функций при использовании метода Монте-Карло значительно меньше, чем при численном интегрировании.

[1] независимые одинаково распределенные случайные величины, имеющие равномерное распределение на отрезке $[a, b]$

2 Необходимые сведения из теории марковских процессов

Случайная последовательность — это такое отображение $X : \Omega \to \mathbb{R}^\infty$, что прообраз $\forall A \in \mathcal{B}(\mathbb{R}^\infty)$ лежит в сигма-алгебре $\mathcal{F}$, то есть $\{\omega : X(\omega) \in A\}$ является элементом $\mathcal{F}$.

Распределение случайной последовательности — это набор вероятностей $P(X \in B)$ для всех $B \in \mathcal{B}(\mathbb{R}^\infty)$.

Конечномерные распределения случайной последовательности — это набор вероятностей $p_{t_1,\dots,t_n}(B_1,\dots,B_n) = P(X_{t_1} \in B_1,\dots,X_{t_n} \in B_n)$ при $t_i \in \mathbb{N}, B_i \in \mathcal{B}(\mathbb{R}^\infty)$. В случае последовательностей целочисленных случайных величин задаём их распределение с помощью набора вероятностей $p_{i_1,\dots,i_n} = P(X_1 = i_1,\dots,X_n = i_n)$.

2.1 Сведения из теории марковских цепей

Случайная последовательность X_n, принимающая значения из конечного или счётного множества S, называется *марковской цепью* [4], если выполняется *марковское свойство*:

$$P(X_n = i_n | X_{n-1} = i_{n-1},\dots,X_0 = i_0) = P(X_n = i_n | X_{n-1} = i_{n-1}) = p_{i_{n-1}i_n}, \tag{5}$$

$\forall\ n = 0,1,\dots$ и $\forall\ i_0,\dots,i_n \in S$ с $P(X_{n-1} = i_{n-1},\dots,X_0 = i_0) > 0$,
то есть любое испытание зависит от предшествующего и только от него.

Если вероятность перехода марковской цепи не зависит от n, то цепь *однородна по времени*, т.е. все переходные вероятности не меняются со временем.

Говорят,что из состояния i *следует* состояние j, если $\exists\ n$:

$$P(X_n = j | X_0 = i) > 0, \tag{6}$$

т.е. находясь в состоянии i возможно оказаться в j.

Матрица $\mathbf{P}$ называется *стохастической*, если $\forall i,j\ 0 \leqslant p_{ij} \leqslant 1$ и $\sum_j p_{ij} = 1\ \forall i$.

Распределение марковской цепи определяют:

- переходная стохастическая матрица цепи $\mathbf{P} = (P(X_t = i | X_{t-1} = j)) = (p_{ji})$,
- первоначальное распределение, т.е. распределение вероятностей в момент времени $t = 0$:

$$\mu = (\mu_1,\dots,\mu_n) = (P(X_0 = i_0)) = (P(X_0 = 1),\dots,P(X_0 = N)), \tag{7}$$

так что $\forall\ n = 0,1,\dots$ и $\forall\ i_0,\dots,i_n \in S$:

$$P(X_0 = i_0,\dots,X_n = i_n) = \mu_{i_0}p_{i_0i_1},\dots,p_{i_{n-1}i_n}. \tag{8}$$

Увидим закономерность в преобразовании распределений.
Рассмотрим распределение на 1-ом шаге:

$$\begin{gathered} p^{(1)} = (p_1^{(1)},\dots,p_n^{(1)}) = (\mu_1 p_{1,1} + \dots + \mu_n p_{n,1}, \mu_1 p_{1,2} + \dots) = \\ (\mu_1,\dots,\mu_n)\begin{pmatrix} p_{1,1} & p_{1,2} & \dots \\ p_{2,1} & \ddots & \\ \vdots & & \end{pmatrix} = \mu\mathbf{P}. \end{gathered} \tag{9}$$

Таким образом, распределение на t-ом шаге:

$$p^{(t)} = (p_1^{(t)},\dots,p_n^{(t)}) = (P(X_t = i_t)) = \mu\mathbf{P}^t. \tag{10}$$

Цепь называется *неразложимой (неприводимой)*, если $\forall i,j$ из i следует j, а из j следует i, иначе говоря, любые состояния $i_a, i_b \in S$ сообщаются, т.е. $\exists n$:

$$P(X_{m+n} = i_a | X_m = i_b) > 0. \tag{11}$$

Рабочий стол/Tex/новая папкachain11.pdf" Рабочий стол/Tex/новая папкachain11.pdf.pdf" Рабочий стол/Tex/новая папкachain11.pdf.png" Рабочий стол/Tex/новая папкachain11.pdf.jpg" Рабочий стол/Tex/новая папкachain11.pdf.mps" Рабочий стол/Tex/новая папкachain11.pdf.jpeg" Рабочий стол/Tex/новая папкachain11.pdf.jbig2" Рабочий стол/Tex/новая папкachain11.pdf.jb2" Рабочий стол/Tex/новая папкachain11.pdf.PDF" Рабочий стол/Tex/новая папкachain11.pdf.PNG" Рабочий стол/Tex/новая папкachain11.pdf.JPG" Рабочий стол/Tex/новая папкachain11.pdf.JPEG" Рабочий стол/Tex/новая папкachain11.pdf.JBIG2" Рабочий стол/Tex/новая папкachain11.pdf.JB2" Рабочий стол/Tex/новая папкachain11.pdf.eps"

Рис. 2. Примеры разложимой (слева) и неразложимой (справа) цепей.

Неразложимая цепь называется *непериодической*, если наибольший общий делитель длин замкнутых путей в этой цепи равен 1. Иначе говоря, есть несколько путей из каких-либо состояний в себя, длины которых взаимно просты.

Рабочий стол/Tex/новая папкachain22.pdf" Рабочий стол/Tex/новая папкachain22.pdf.pdf" Рабочий стол/Tex/новая папкachain22.pdf.png" Рабочий стол/Tex/новая папкachain22.pdf.jpg" Рабочий стол/Tex/новая папкachain22.pdf.mps" Рабочий стол/Tex/новая папкachain22.pdf.jpeg" Рабочий стол/Tex/новая папкachain22.pdf.jbig2" Рабочий стол/Tex/новая папкachain22.pdf.jb2" Рабочий стол/Tex/новая папкachain22.pdf.PDF" Рабочий стол/Tex/новая папкachain22.pdf.PNG" Рабочий стол/Tex/новая папкachain22.pdf.JPG" Рабочий стол/Tex/новая папкachain22.pdf.JPEG" Рабочий стол/Tex/новая папкachain22.pdf.JBIG2" Рабочий стол/Tex/новая папкachain22.pdf.JB2" Рабочий стол/Tex/новая папкachain22.pdf.eps"

Рис. 3. Примеры периодичных цепей: слева с периодом k = 2, справа с k = 3.

Сведения, изложенные в пункте 2.1 будут использованы в следующем разделе.

2.2 Эргодическая теорема

Рассмотрим однородную марковскую цепь с двумя состояниями 0 и 1, которая задается матрицей переходных вероятностей

$$\mathbf{P} = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix}. \tag{12}$$

Если в выражении (10) положить $t = n$, то по индукции получаем

$$\mathbf{P}^n = \frac{1}{2 - p_{11} - p_{22}} \begin{pmatrix} 1 - p_{22} & 1 - p_{11} \\ 1 - p_{22} & 1 - p_{11} \end{pmatrix} + \frac{(p_{11} + p_{22} - 1)^n}{2 - p_{11} - p_{22}} \begin{pmatrix} 1 - p_{11} & -(1 - p_{11}) \\ -(1 - p_{22}) & 1 - p_{11} \end{pmatrix}. \tag{13}$$

В выражении (13) устремим $n \to \infty$, тогда

$$\mathbf{P}^n \to \frac{1}{2 - p_{11} - p_{22}} \begin{pmatrix} 1 - p_{22} & 1 - p_{11} \\ 1 - p_{22} & 1 - p_{11} \end{pmatrix}. \tag{14}$$

Таким образом, с течением времени $p_{ji}^{(n)}$ сходятся к предельным значениям π_i, не зависящим от j [7].

Далее приведем теорему из работы [7].

Теорема 1. *Если конечная цепь Маркова с переходной матрицей* $\mathbf{P}$ *является неразложимой и апериодичной, то при любом начальном распределении* λ *вероятности* $P(X_n = i)$ *сходятся к некоторым* π_i, $i = 1, \ldots, N$ *при* $n \to \infty$*:*

$$\mathbf{P}^n \to \ \mathbf{\Pi}, n \to \infty. \tag{15}$$

При этом вектор $\pi = (\pi_1, \ldots, \pi_N)$ *является единственным решением системы уравнений*

$$\pi \mathbf{P} = \pi \tag{16}$$

и называется эргодическим (стационарным, инвариантным) распределением.

Таким образом, для неприводимой апериодической цепи Маркова с течением времени получаем распределение, не зависящее от первоначального:

$$\lim_{n \to \infty} P(X_n = j) = \pi_i \ \forall \lambda. \tag{17}$$

Следующая теорема с доказательством приведена из [5].

Теорема 2. *Пусть* X_n *— цепь Маркова. Следующие свойства эквивалентны:*

1. *Цепь обратима, т.е.* $\forall n \geqslant 1$ *и* $\forall$ *состояний* $i_0, \ldots, i_n$

$$P(X_0 = i_0, \ldots, X_n = i_n) = P(X_0 = i_n, \ldots, X_n = i_0); \tag{18}$$

2. *Цепь маркова* X_n *находится в состоянии равновесия* $(X_n \sim (\pi, \mathbf{P}))$*, где* π *— стационарное распределение для* $\mathbf{P}$*, или, другими словами,*
выполнены уранения детального баланса (сбалансированности) $\forall$ *состояний* i, j*, если:*

$$\pi_i p_{ij} = \pi_j p_{ji}. \tag{19}$$

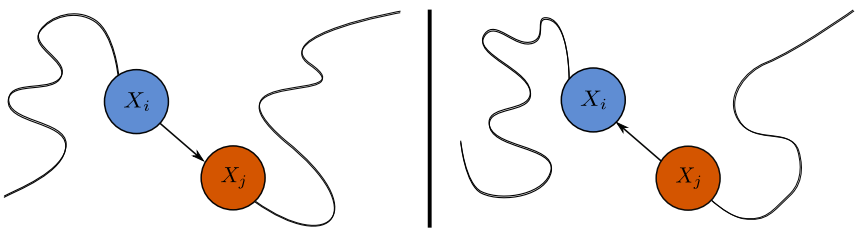

Рис. 4. Два состояния обратимой марковской цепи.

Доказательство. Действительно, рассмотрим цепь с матрицей перехода $\mathbf{P}$ и начальным распределением λ:
1) $\Rightarrow$ 2). Пусть $n = 1$:

$$P(X_0 = i, X_1 = j) = P(X_0 = j, X_1 = i). \tag{20}$$

Просуммируем по j:

$$\sum_j P(X_0 = i, X_1 = j) = P(X_0 = i) = \lambda_i, \tag{21}$$

$$\sum_j P(X_0 = j, X_1 = i) = P(X_1 = i) = (\lambda\mathbf{P})_i. \tag{22}$$

Таким образом, $\forall i\ \lambda_i = (\lambda\mathbf{P})_i$, т.е. $\lambda = (\lambda\mathbf{P}) \Rightarrow$ цепь находится в состоянии равновесия со стационарным распределением λ. Далее,

$$\begin{gathered} P(X_0 = i, X_1 = j) = P(X_1 = j|X_0 = i)P(X_0 = i) = p_{ji}\lambda_i = \\ P(X_0 = j, X_1 = i) = P(X_1 = i|X_0 = j)P(X_0 = j) = p_{ij}\lambda_j, \end{gathered} \tag{23}$$

т.е. имеет место условие сбалансированности. 2) $\Rightarrow$ 1). Перепишем

$$P(X_0 = i_0, \ldots, X_n = i_n) = \pi_{i_0} p_{i_0 i_1} \ldots p_{i_{n-1} i_n} \tag{24}$$

и воспользуемся уравнением (11):

$$\begin{gathered} \pi_{i_0} p_{i_0 i_1} \ldots p_{i_{n-1} i_n} = \pi_{i_1} p_{i_1 i_0} \ldots p_{i_{n-1} i_n} = p_{i_1 i_0} \pi_{i_1} p_{i_1 i_2} \ldots p_{i_{n-1} i_n} = \\ p_{i_1 i_0} \pi_{i_2} p_{i_2 i_1} \ldots p_{i_{n-1} i_n} = \cdots = p_{i_1 i_0} p_{i_2 i_1} \ldots \pi_{i_n} p_{i_n i_{n-1}} = \pi_{i_n} p_{i_n i_{n-1}} \ldots p_{i_1 i_0} = \\ P(X_0 = i_n, \ldots, X_n = i_0) \blacktriangleright \end{gathered} \tag{25}$$

Условие сбалансированности является сильным инструментом для нахождения стационарного распределения:

Теорема 3. *Если λ и $\mathbf{P}$ удовлетворяют уровнениям детального баланса*

$$\lambda_i p_{ij} = \lambda_j p_{ji}, \tag{26}$$

то λ является стационарным распределением для $\mathbf{P}$, т.е. $\lambda\mathbf{P} = \lambda$.

Доказательство. Просуммируем по j:

$$\sum_j \lambda_i p_{ij} = \lambda_i \sum_j p_{ij} = \lambda_i, \tag{27}$$

$$\sum_j \lambda_j p_{ji} = (\lambda\mathbf{P})_i. \tag{28}$$

$\forall i$ выражения равны $\Rightarrow \lambda$ — стационарное распределение и цепь обратима по теореме 2.$\blacktriangleright$

Таким образом, если для заданного распределения p удастся найти такую матрицу перехода $\mathbf{P}$, что будет выполнено условие сбалансированности, то это распределение будет сходиться к стационарному с течением времени.

2.3 Марковские цепи с непрерывным множеством состояний

Основные идеи Марковской цепи с дискретным пространством состояний могут быть обобщены на непрерывный случай.

Марковским процессом X_n с дискретным временем и множеством (уже не обязательно не более чем счетным) состояний S будем называть такую последовательность, что

$$P(X_n \in A | X_{n-1} = x_{n-1}, \dots, X_0 = x_0) = P(X_n \in A | X_{n-1} = x_{n-1}). \tag{29}$$

Здесь $P(X_n \in A | X_{n-1}) = E(I_{X_n \in A} | X_{n-1} = x_{n-1})$ — условное математическое ожидание.

Функция $P(X_n \in A | X_{n-1} = x_{n-1}) = p(x_{n-1}, A, n)$ называется переходной функцией марковского процесса, если $S \subset \mathbb{R}$. Если существует плотность $f_{X_n|X_{n-1}}(y|x)$, то она называется переходной плотностью $P_{x,y,n}$.

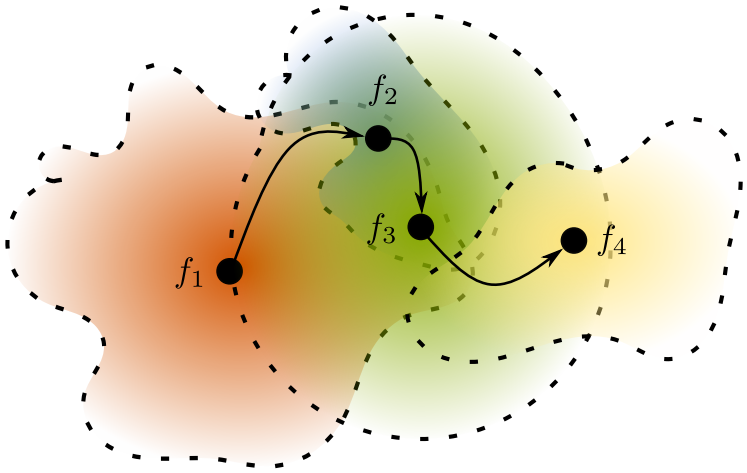

Рис. 5. Визуализация марковского процесса: новое состояние зависит от плотности распределения ему предшествующего.

Рассмотрим стационарное распределение π с переходной плотностью P, теперь уравнение Колмогорова - Чепмена [4] примет вид:

$$\int_x \pi(x) P(x, y) dx = \pi(y). \tag{30}$$

Обратимой назовем цепь, для которой выполняется уравнение детального баланса:

$$P(x, y) p(x) = P(y, x) p(y). \tag{31}$$

Опять условие сбалансированности влечет стационарность распределения.

Из эргодической теоремы для случая с непрерывным множеством состояний [8] мы получаем, что распределение процесса сходится к стационарному.

Тем самым, метод Монте-Карло будет работать и в случае марковских процессов с континуальным множеством состояний.

Общая идея метода Монте-Карло Марковских цепей (MCMC):

- построить цепь Маркова с эргодическим распределением, для которой стационарное распределение - это в точности заданное нами распределение,
- инициировать блуждание по цепи Маркова из некоторого начального состояния и дождаться, когда распределение сойдется к стационарному. С этого момента состояния цепи Маркова можно считать выборкой из желаемого распределения.

3 Гиббсовское семплирование

3.1 Идея и математическое обоснование

В некоторых типах задач одномерные условные распределения гораздо легче моделировать, чем совместные распределения. Идея гиббсовского семплирования заключается в том, что для восстановления совместного распределения рассматриваются только условные распределения для каждой переменной.
Для наглядности, рассмотрим двумерную случайную величину (x, y). Мы хотим вычислить маргинальные распределения $p(x)$ и $p(y)$. Проще рассмотреть условные распределения $p(x|y)$, $p(y|x)$, чем искать совместную плотность посредством интегрирования $p(x) = \int p(x, y)dy$.
Семплирование начинается с некоторого значения y_0 для y. Генерируя случайную величину из условного распределения $p(x|y = y_0)$ получаем x_0. Далее используем x_0 для генерации нового значения y_1 из условного распределения $p(y|x = x_0)$. И так далее:

$$x_i \sim p(x|y = y_{i-1}),\ y_i \sim p(y|x = x_i). \tag{32}$$

Повторяя этот процесс k раз, получаем последовательность Гиббса длиной k с элементами (x_j, y_j), $1 \leqslant j \leqslant k$. Последовательность Гиббса сходится к стационарному распределению, которое и является искомым.

Рассмотрим случай нескольких переменных: пусть явный вид распределения вычислить трудно, но известны условные плотности

$$f_{X_i|X_1,\ldots,X_{i-1},X_{i+1},\ldots,X_n}(x_i|x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n). \tag{33}$$

Нам нужно сгенерировать вектор $(X_1, \ldots, X_n)$, где мы знаем условные плотности.

- Выберем некоторый начальный вектор $(X_{1,1}, \ldots, X_{1,n})$, который вообще может получиться из нашего распределения.
- Выберем случайный индекс d из множества $1, \ldots, n$.
- Положим $X_{2,i} = X_{1,i}$ при $i \neq d$. Величину $X_{2,d}$ сделаем случайной с плотностью $f_{X_d|X_1,\ldots,X_{d-1},X_{d+1},\ldots,X_n}(x_d|x_1,\ldots,x_{d-1},x_{d+1},\ldots,x_n)$.
- Повторим шаг 2,3 для нового вектора и так далее.

Вектор $(X_{m,1}, \ldots, X_{m,n})$ сходится по распределению к требуемому вектору $(X_1, \ldots, X_n)$ при $m \to \infty$.

Алгоритм на каждом шаге берет случайную величину и задает ее значение при фиксированных остальных. Таким образом, последовательно моделируются n случайных величин из n одномерных условных выражений вместо того, чтобы генерировать один n-мерный вектор за один подход с использованием полного совместного распределения. Последовательность генерируемых значений образует обратимую цепь маркова, эргодическое распределение которой является искомым.

Убедимся, что алгоритм гиббсовского семплирования укладывается в изложенную теорию.
В данном случае мы рассматриваем процесс с переходной плотностью из $\vec{x} = (x_1, \ldots, x_n)$ в $\vec{y} = (x_1, \ldots, x_{k-1}, z, x_{k+1}, \ldots, x_n)$.

$$P(\vec{x}, \vec{y}) = \frac{1}{n} f_{X_k|X_1,\ldots,X_{k-1},X_{k+1},\ldots,X_n}(z|x_1,\ldots,x_{k-1},x_{k+1},\ldots,x_n)\,. \tag{34}$$

Проверим, что она удовлетворяет условию сбалансированности с нужной нам $p(\vec{x}) = f_{X_1,\ldots,X_n}(\vec{x})$. Действительно,

$$P(\vec{x}, \vec{y})p(\vec{x}) = \frac{1}{n} \frac{f_{X_1\ldots,X_n}(x_1\ldots,x_{k-1},z,x_{k+1},\ldots,x_n)}{f_{X_1\ldots X_{k-1},X_{k+1},\ldots,X_n}(x_1,\ldots,x_{k-1},x_{k+1},\ldots,x_n)} f_{X_1,\ldots,X_n}(x_1,.,x_n)\,. \tag{35}$$

При этом $P(\vec{y}, \vec{x})p(\vec{y})$ даст ту же формулу с точностью до перестановки множителей. Следовательно, построенная цепь является сбалансированной с нужной плотностью.

3.2 Задача про одинаково распределенные точки в круге

Предположим, что мы хотим взять наугад выборку из N точек в единичном круге, удаленных друг от друга не менее чем на расстояние d. Выписать для данной задачи совместную плотность вектора трудно, однако условные плотности устроены довольно просто. Если мы знаем все точки, кроме одной, то оставшаяся точка распределена равномерно на исходном круге за вычетом кругов радиуса d вокруг остальных точек.

Следуя алгоритму гиббсовского семплирования :

- На первом шаге мы случайным образом выбираем точку в исходном круге радиуса 1 U_1. Назовем эту точку X_1. Пусть $U_d(X_1)$—круг радиуса d с центром в точке X_1. Вторая точка выбирается случайно из множества $U_1 \setminus U_d(X_1)$. Третья точка X_3 выбирается случайно из множества $U_1 \setminus U_d(X_1) \setminus U_d(X_2)$. И так далее.
- Дальше мы генерируем случайный индекс i и меняем элемент X_i нашей выборки $(X_1, \ldots, X_N)$ на точку, равномерно распределённую в единичном круге без остальных $N-1$ кругов радиусов d. После большого числа итераций полученный набор из N точек будет иметь искомое распределение.

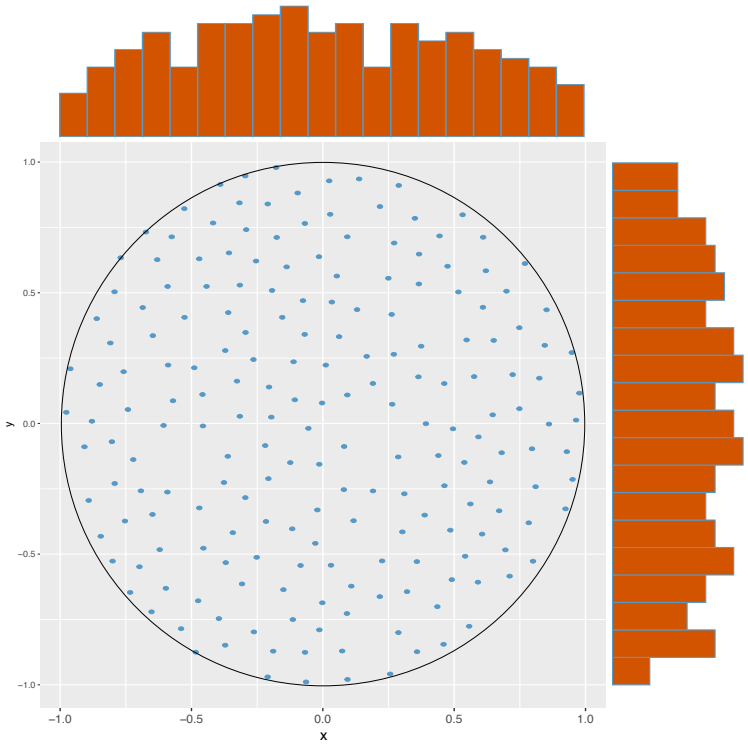

Рис. 6. Равномерное распределение точек в круге.

Если мы не будем использовать гиббсовское семплирование, то уже вторая сгенерированная точка не будет попадать в центр круга, поскольку её будет вытеснять равномерно - распределённая первая точка. Значит, первая и вторая точки не будут одинаково распределены.

Чтобы это заметить, пронаблюдаем распределение первой и второй брошенных в круг точек при гиббсовском семплировании и без (рис.7. и рис. 8.).

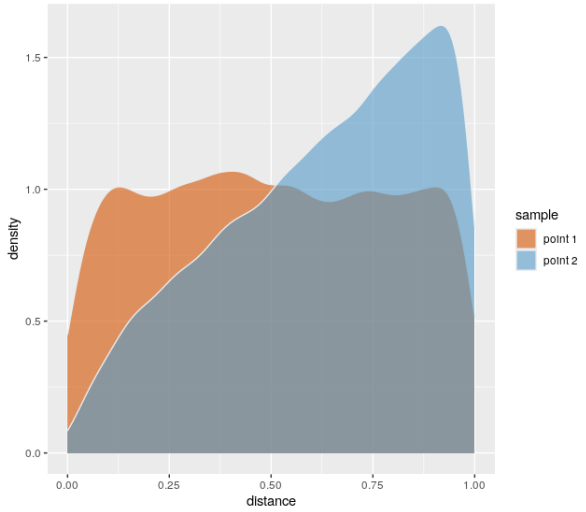

Рис. 7. Распределения точек без семпирования.

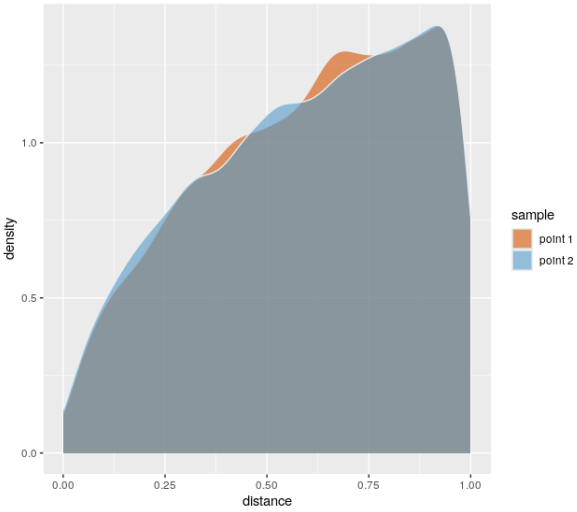

Рис. 8. Распределения с семплированием.

Для данной работы алгоритм гиббсовского семплирования и графики были реализованы с помощью языка программирования *R*. Код доступен по ссылке [9]. Иллюстрации выполнены в графическом редакторе Inkscape.

Вывод

Мы показали, что можно построить марковскую цепь с заранее заданным эргодическим распределением. В эту теорию укладывается метод гиббсовского семплирования, который позволяет моделировать совместное распределение, используя только условные, что облегчает вычисления во многих задачах. Например, в рассмотренной задаче заполнения круга одинаково распределенными точками использование гиббсовского семплирования позволяет восстановить искомое распределение, исходя из условных плотностей. Задача восстановления данных имеет ту же структуру: совместное распределение посчитать сложно, но условные распределения приблизительно известны. Таким образом, изученный алгоритм может быть использован для решения задачи восстановления данных.

Приложения

```
library(hexbin)
library(RColorBrewer)
library(ggplot2)
library(ggExtra)

rcircle = function(r=1){
  r = r * sqrt(runif(1))
  theta = runif(1) * 2 * pi

  x = r * cos(theta)
  y = r * sin(theta)
  return(c(x,y))
}

start <- function(N_points, r, d){
  start_samp <- matrix(-r*100, N_points, 2) #Generate points far away from center

  A = rcircle(r)
  x1 <- A[1]
  y1 <- A[2]
  i = 1
  j = 1
  k = 0 #flag for searching the right second point
  c=0

  for (i in 1:N_points){
    #the second point's generation
    while (k == 0){
      A = rcircle(r)
      x2 <- A[1]
      y2 <- A[2]

      c = c+1
      #If we cant generate point for 10 000 times - break
      if (c == 10000){
        warning("Generation timed out")
        return()
      }

      #the distance between current point and all that we've already taken
      for (j in 1:N_points){
        x_sq = (start_samp[j,1] - x2)^2
        y_sq = (start_samp[j,2] - y2)^2
        distance = sqrt(x_sq + y_sq)

        if (distance < d){
          k = 0
          break
        }
        #if we didn't get into the previous 'if' conditions => our 2nd point is correct and then j
        #if we got into this conditions then 'break' operator will stop it and there will be selec
        k = 1
      }
    }
```

```
    start_samp[c(i), 1] <- x2
    start_samp[c(i), 2] <- y2
    k = 0
  }
  return(start_samp)
}

sampling <- function(start_samp, r, d, N_samling = 100, silent = T){
  j = 1
  k = 0
  m = 1
  c=0

  N_points = NROW(start_samp)
  if (!silent)
    pb = txtProgressBar(min = 0, max = N_samling, style = 3) #fashion progressbar
  for (m in 1:N_samling){
    if (!silent)
      setTxtProgressBar(pb,m)

     current = sample(1:N_points, 1) #random index

    #the second point's ganeration
    while (k == 0){

      A = rcircle(r)
      xi <- A[1]
      yi <- A[2]

      #the distance between current point and all that we've already taken
      for (j in 1:N_points){
        if ( j != current){

          x_sq = (start_samp[j,1] - xi)^2
          y_sq = (start_samp[j,2] - yi)^2
          distance = sqrt(x_sq + y_sq)

          if ( distance < d){
            k = 0
            break
          }
          k = 1
        }
      }
    }
    start_samp[current, 1] = xi
    start_samp[current, 2] = yi
    k = 0
  }
  if (!silent)
    close(pb)

  return(start_samp)
}

D = 0.9
```

```
R = 1
N_POINTS = 2
N_SAMPLING = 1e4

  A = start(N_points = N_POINTS, r = R, d = D)
  plot(A[,1], A[,2], pch=16, ylim=c(-1,1), xlim=c(-1,1))

  A_resampled= sampling(A, r = R, d = D, N_samling = N_SAMPLING)

#-------------------------
  plot(A_resampled[,1], A_resampled[,2], pch=16)
  library(ggplot2)
  library(ggExtra)

  d1 <- as.data.frame(A_resampled, stringsAsFactors=FALSE)
  x <- d1[,1]
  y <- d1[,2]

  p <- ggplot(d1, aes(x=x, y=y)) +
    geom_point(fill="slateblue",col="#5499c7") +
    theme(legend.position="none")
  p
  # with marginal histogram
  p1 <- ggMarginal(p, type="histogram", fill = "#d35400",col = "#5499c7", size = 3, xparams = list
  p1
#------------------
  STOP = 1e4
  D = 1
  R = 1
  N = 2
  Nsamp = 100

  nosamp_matr =start(N_points = N_POINTS, r = R, d = D)
  samp_matr = sampling(A,r = R, d = D, N_samling = Nsamp)

  PB = txtProgressBar(min = 0, max = STOP, style = 3)

  for (i in 1:(STOP - 1)){
    setTxtProgressBar(PB, i)

    A = start(N_points = N, r = R, d = D)
    nosamp_matr = cbind(nosamp_matr, A)
    B = sampling(A,r = R, d = D, N_samling = Nsamp)
    samp_matr = cbind(samp_matr, B)
  }

  fir = matrix(samp_matr[1,], byrow = T, ncol = 2)
  sec = matrix(samp_matr[2,], byrow = T, ncol = 2)

  fir_NO = matrix(nosamp_matr[1,], byrow = T, ncol = 2)
  sec_NO = matrix(nosamp_matr[2,], byrow = T, ncol = 2)

  dist1 = data.frame(abs(fir[,1]^2+fir[,2]^2))
  dist2 = data.frame(abs(sec[,1]^2+sec[,2]^2))
  names(dist1) = "dist"
  names(dist2) = "dist"
```

```
dist1$name = "dist1"
dist2$name = "dist2"

df = rbind(dist1, dist2)
names(df) = c("dist", "sample")
ggplot(df, aes(dist, fill = sample)) + geom_histogram(color="#e9ecef",alpha = 0.6, position = "i
ggplot(df, aes(dist, fill = sample)) + geom_density(color="#e9ecef",alpha = 0.6)+scale_fill_manu

dist1_NO = data.frame(abs(fir_NO[,1]^2+fir_NO[,2]^2))
dist2_NO = data.frame(abs(sec_NO[,1]^2+sec_NO[,2]^2))
names(dist1_NO) = "dist"
names(dist2_NO) = "dist"
dist1_NO$name = "dist1"
dist2_NO$name = "dist2"

df = rbind(dist1_NO, dist2_NO)
names(df) = c("dist", "sample")
ggplot(df, aes(dist, fill = sample)) + geom_histogram(color="#e9ecef",alpha = 0.6, position = "i
ggplot(df, aes(dist, fill = sample)) + geom_density(color="#e9ecef",alpha = 0.6)+scale_fill_manu
```

Список литературы

[1] Муромцева, Г. А., et al. Распространенность факторов риска неинфекционных заболеваний в российской популяции в 2012-2013гг. Результаты исследования ЭССЕ-РФ. Кардиоваскулярная терапия и профилактика 13(6) (2014).

[2] Buuren, S. V., Groothuis-Oudshoorn, K. Mice: Multivariate imputation by chained equations in R. Journal of statistical software (2010).

[3] Metropolis, N., Ulam, S. The monte carlo method. Journal of the American statistical association, 44(247) (1949).

[4] Феллер, В. Введение в теорию вероятностей и ее приложения (Vol. 2). Рипол Классик (2013).

[5] Кельберт, М., Сухов, Ю. Вероятность и статистика в примерах и задачах. Том 2. Марковские цепи как отправная точка теории случайных процессов и их приложения. Litres (2017).

[6] Murphy, K. P. Machine learning: a probabilistic perspective. MIT press (2012).

[7] Ширяев, А. Н. Вероятность. В 2-х кн. Москва: МЦНМО (2004).

[8] Bellet, L. R. Ergodic properties of Markov processes. In Open quantum systems II (pp. 1-39). Springer, Berlin, Heidelberg (2006).

[9] `https://github.com/annalimm/Gibbs_circle.git`.